Project proposal for the mobile application game.

CS 596 – Fall 2014



Happy Fishing

## Prepared by

## Krishna Matta

## Brandon fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Author** | **Changes** |
| 09/23/2014 | 1.0 | Brandon Fields | Initial Preparing, Description |
| 09/26/2014 | 1.1 | Krishna Matta | Drawing, Implementation description, milestone |
| 09/29/2014 | 1.2 | Brandon Fields | Added software design plan |
| 11/19/14 | 1.3 | Brandon Fields | Updated Game Description and Work Division. Added Challenges |

# Game Description

The game is a spear fishing game with multiple levels. The player will use a spear, located on the screen, to aim and shoot at fish. There will be a variety of types of fish, each being worth a predetermined amount of points. There will also be obstacles that the player should avoid hitting. Hitting obstacles will result in the loss of points in your total score. If the player reaches the point threshold for that level, he or she will advance to the next level. Each level increases in difficulty. The level difficulty is interpreted by an increasing number of obstacles, more score required per spears. The player will experience an underwater view. Each background image will change as the player successfully progresses through levels.

Some basic implementation:

* Start the game.
* Display the scene with moving fishes and any moving or static obstacles.
* Displays the score board with points, level, and number of spears left.
* Displays the spear bottom center of the screen.
* Player will hold and drag the end point of spear to set the direction of the spear.
* Player hold and pulls back to apply force in the forward direction. The more he/she pulls, the more force the spear gets.
* Player will hold and drag the spear hinge to change position of the spear. Hinge of spear movement restricted to particular area, like hinge cannot moved to water or score board.(this is not implemented as this feature is not useful)
* If the spear hits (collides) the fish, fish will be disappeared and score will be displayed with some kind of animation, player gets the appropriate points.
* If the spear hits any obstacles, player loses points and/or life.
* If the player gets certain points for a level he will be promoted to next level, otherwise player get a chance to play the current level.
* For each level, level target score will be incremented, and more obstacles will be added.
* Successful final level completion shows, score of the final level, levels completion message.

# Target platform

We have compiled, installed and tested this game for both android (minimum SDK 2.3.3) and iOS devices.

# Intended audience

Everyone (children and adults)

# Screen Shots

1. Home Screen



1. Level 1 screen



1. Level 1 with some score



1. Shooting with spear



1. Level pause overlay display



1. Level Failed overlay



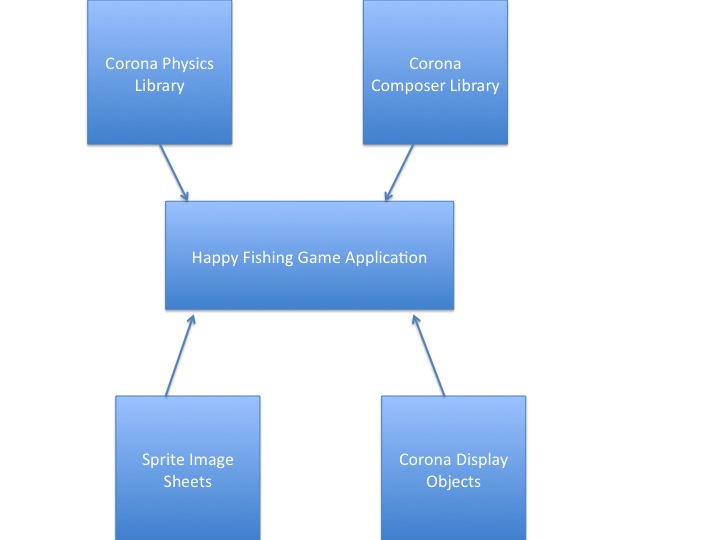
1. Level completion overlay



1. Final level completion



# Discussion of software design plan



The software design will utilize four major components.

* **Corona Physics Library** – This library will be used for a simulation of objects that move, collide, and interact under various physical forces like gravity. The main objects will be the fish, obstacles, and spear. Corona physics engine works with Corona [display objects](http://docs.coronalabs.com/daily/guide/media/displayObjects/index.html) and Sprite images.
* **Corona Composer Library** – This library is the screen creation and management library. We will use this library to set up the game screen as well as transition to different levels throughout the game.
* **Sprite Image Sheets**- We will use these sheets to load multiple images from a single user file. Sprite Image Sheets will allow the developers to minimize the amount of memory needed to store the static or dynamic animation objects.
* **Corona Display Objects** – These are the objects for basic shapes that the Corona Library provides. We will use the shapes to create the scoreboard as well as other information that is needed to be displayed to the screen.

# Implemented Design and all components

main.lua

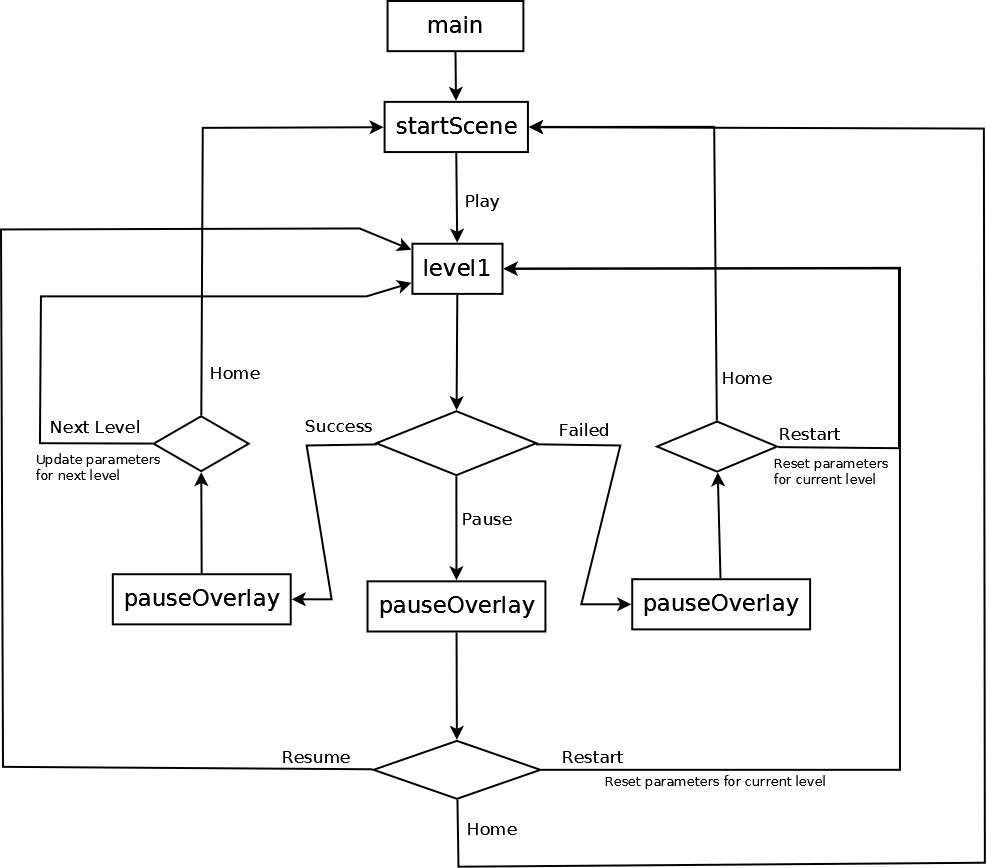
startScene.lua

level1.lua

pauseOverlay.lua

reloading.lua

1. Game Workflow



# Work division amongst group members

1. Design implementation. (Krishna and Brandon)
2. Image and characters preparations. (Krishna and Brandon)
3. Score box design (Krishna)
4. Spear design and implement. (Krishna)
5. Animations and physics. (Krishna and Brandon)
6. Level designs. (Krishna)
7. Overlay Scenes (Brandon)
8. Present end scene and scores. (Krishna and Brandon)

# Challenges

1. Positioning the spear on the screen.

* We originally positioned the spear in the bottom right corner of the screen. This seemed fine until we tested on real hardware. The lower right corner position made the spear difficult to shoot. We decided to move the spear to the lower middle of the screen.

1. Manually creating new levels

* Since we decided to go with and OO approach it was feasible for new levels to be created dynamically. We have a field that we can set that will limit the number of levels that is created.

1. Handling the touch event for Spear:

* Initially added touch event for the spear for the rotate and shoot the enemy. As simulator works with mouse, we didn’t notice any issues in shooting. But when we created the .apk and tested in the real device, we noticed that selecting the spear is very difficult as it is very thin. So we decided to add the touch event for the Base circle and simulated such that, touching the base will update the spear angle and leaving touch, applies the force.
* When we drag the finger on the base, and if the finger leaves the circle, touch event was ending at that point and shooting the spear. To solve this issue, at the beginning of the touch setting the base object as the focus object, so that even the touch point leaves the circle, still touch event will fire, until we lift the finger from the screen.

1. Finding the required images and buttons:

* Initially we have created images very small and when we test on the device, it was very hard to play and annoying. So made the images little bit bigger. Used Piant.Net software to create sprite sheet for images and buttons. Hover state buttons created manually from the normal state buttons.

1. Build issues:

* Initially game was not orienting to match the device orientation,

Added *supported = { "landscapeLeft", "landscapeRight" },* to orientation settings

# Project milestones and schedule

Mile stone 1: Sept 30, 2014 - Project proposal.

Mile stone 2: Oct 16, 2014 - Basic design implementation with characters.

Mile stone 3: Oct 30, 2014 - Score board design and spear design.

Mile stone 4: Nov 13, 2014 - Animation and physics and collision detection implementation.

Mile stone 5: Dec 02, 2014 - Levels implementation, testing and Submit.

# References

* 1. Pool games to shoot balls.
  2. Course slides and assignments.
  3. Buttons downloaded from

<http://findicons.com>

* 1. sound files downloaded from

<http://soundbible.com/>

* 1. Fish images downloaded from

<https://openclipart.org/search/?query=fish&page=1>